

In response to execution of the function call by client 52, the application 17 transmits to the master pipeline 55 a command to render the X window 245 and a command to render a 3D image within portion 249 of the X window 245. The command for rendering the X window 245 should include 2D graphical data defining the X window 245, and the  
5 command for rendering the 3D image within the X window 245 should include 3D graphical data defining the 3D image to be displayed within region 249. Preferably, the master pipeline 55 renders 2D graphical data from the former command (*i.e.*, the command for rendering the X window 245) to frame buffer 65 (FIG. 3) via X server 162 (FIG. 6).

The graphical data rendered by any of the pipelines 55-59 includes sets of values that  
10 respectively define a plurality of pixels. Each set of values includes at least a color value and a plurality of coordinate values associated with the pixel being defined by the set of values. The coordinate values define the pixel's position relative to the other pixels defined by the graphical data, and the color value indicates how the pixel should be colored. While the coordinate values indicate the pixel's position relative to the other pixels defined by the  
15 graphical data, the coordinate values produced by the application 17 are not the same coordinate values assigned by the display device 83 to each pixel of the screen 247. Thus, the pipelines 55-59 should translate the coordinate values of each pixel rendered by the pipelines 55-59 to the coordinate values used by the display device 83 to display images. Sometimes the coordinate values produced by the application 17 are said to be "window  
20 relative," and the aforementioned coordinate values translated from the window relative coordinates are said to be "screen relative." The concept of translating window relative coordinates to screen relative coordinates is well known, and techniques for translating window relative coordinates to screen relative coordinates are employed by most conventional graphical display systems.

In addition to translating coordinates of the 2D data rendered by the master pipeline 55 from window relative to screen relative, the master pipeline 55 in each mode of operation also assigns a particular color value, referred to hereafter as the “chroma-key,” to each pixel within the region 249. The chroma-key indicates which pixels within the X window 245 may be assigned a color value of a 3D image that is generated by slave pipelines 56-59. In this regard, each pixel assigned the chroma-key as the color value by master server 55 is within region 249 and, therefore, may be assigned a color of a 3D object rendered by slave pipelines 56-59, as will be described in further detail hereafter. In the example shown by FIG. 7, the graphical data rendered by master pipeline 55 and associated with screen relative coordinate values ranging from (700, 700) to (1300, 1300) are assigned the chroma-key as their color value by the master pipeline 55, since the region 249 is the portion of X window 245 that is to be used for displaying 3D images.

As shown by FIG. 5, the master pipeline 55 includes a slave controller 261 that is configured to provide inputs to each slave pipeline 56-59 over the LAN 62. The slave controller 261 may be implemented in software, hardware, or a combination thereof, and in the embodiment shown by FIG. 5, the slave controller 261 is implemented in software and stored in memory 206. The inputs from the slave controller 261 inform the slaves 56-59 of which mode each slave 56-59 should presently operate. In the present example, the slave controller 261 transmits inputs to each slave 56-59 indicating that each slave 56-59 should be in the optimization mode of operation. The inputs from slave controller 261 also indicate which portion of region 249 (FIG. 7) that is each slave’s responsibility. For example, assume for illustrative purposes, that each slave 56-59 is responsible for rendering the graphical data displayed in one of the portions 266-269 shown by FIG. 8.

In this regard, assume that: (1) slave pipeline 56 is responsible for rendering graphical data defining the image displayed in portion 266 (*i.e.*, screen relative coordinates

(700, 1000) to (1000, 1300), (2) slave pipeline 57 is responsible for rendering graphical data defining the image displayed in portion 267 (*i.e.*, screen relative coordinates (1000, 1000) to (1300, 1300)), (3) slave pipeline 58 is responsible for rendering graphical data defining the image displayed in portion 268 (*i.e.*, screen relative coordinates (700, 700) to (1000, 1000)), and (4) slave pipeline 59 is responsible for rendering graphical data defining the image displayed in portion 269 (*i.e.*, screen relative coordinates (1000, 700) to (1300, 1000)). The inputs transmitted by the slave controller 261 to the slave pipelines 56-59 preferably indicate the range of screen coordinate values that each slave pipeline 56-59 is responsible for rendering.

Note that the partition of the region 249 can be divided among the pipelines 56-59 via other configurations, and it is not necessary for each pipeline 56-59 to be responsible for an equally sized area of the region 249. For example, FIG. 9 shows an embodiment where each portion 266-269 represents a different sized horizontal area of the region 249.

Each slave pipeline 56-59 is configured to receive from master pipeline 55 the graphical data of the command for rendering the 3D image to be displayed in region 249 and to render this data to frame buffers 66-69, respectively. In this regard, each pipeline 56-59 renders graphical data defining a 2D X window that displays a 3D image within the window. More specifically, slave pipeline 56 renders graphical data to frame buffer 66 that defines an X window displaying a 3D image within portion 266 (FIG. 8). The X server 202 within slave pipeline 56 renders the data that defines the foregoing X window, and the OGL daemon 205 within the slave pipeline 56 renders the data that defines the 3D image displayed within the foregoing X window. Furthermore, slave pipeline 57 renders graphical data to frame buffer 67 that defines an X window displaying a 3D image within portion 267 (FIG. 8). The X server 202 within slave pipeline 57 renders the data that defines the foregoing X window, and the OGL daemon 205 within the slave pipeline 57 renders the